Uniting contrastive and generative learning for event sequences models

Aleksandr Yugay and Alexey Zaytsev

Skolkovo Institute of Science and Technology, Moscow, Russian Federation
{Aleksandr.Yugay, A.Zaytsev}@skoltech.ru

Abstract. High-quality representation of transactional sequences is vital for modern banking applications, including risk management, churn prediction, and personalized customer offers. Different tasks require distinct representation properties: local tasks benefit from capturing the client's current state, while global tasks rely on general behavioral patterns. Previous research has demonstrated that various self-supervised approaches yield representations that better capture either global or local qualities.
This study investigates the integration of two self-supervised learning techniques — instance-wise contrastive learning and a generative approach based on restoring masked events in latent space. The combined approach creates representations that balance local and global transactional data characteristics. Experiments conducted on several public datasets, focusing on sequence classification and next-event type prediction, show that the integrated method achieves superior performance compared to individual approaches and demonstrates synergistic effects. These findings suggest that the proposed approach offers a robust framework for advancing event sequences representation learning in the financial sector.

Keywords: self-supervised learning · contrastive learning · generative learning · event sequences · transactional data.

1 Introduction

Transforming vast amounts of information into actionable insights enhances decision-making, optimizes operations, and improves customer experiences - this observation holds in the banking industry as well [1, 18]. Machine learning (ML) plays a critical role in this transformation, allowing financial institutions to handle complex tasks and extract valuable insights from their data [4].

Using models such as deep networks and hybrid models, banks accurately evaluate credit scores, predict financial distress, and detect risky transactions. This enhances their ability to assess risks and make informed lending decisions for a wide range of problems [8].

Recent developments in ML mostly concern the adoption of neural networks for financial data, including financial transactions modality [15]. One of the reasons for their wide adoption is representation learning capabilities. Data embedding is the core aspect of representation learning methods. It transforms raw

data points or sequences into low-dimensional, fixed-length vectors. These vectors are designed to capture the underlying patterns or "nature" of the data, making them valuable for a wide range of downstream tasks [7].

Pre-trained embeddings from different domains are frequently employed for various purposes [14, 20]. They can be used as informative, ready-to-use input features for ML or deep learning models, reducing the need for extensive domain knowledge or significant feature engineering. Alternatively, these embeddings can serve as foundational building blocks when dealing with composite multi-modal data, where data from various sources or types are combined. The most powerful methods here consider the self-supervised learning paradigm, where we train the encoder using unlabeled data [20].

This idea is also adopted for financial transactions data [2, 7]. For example, [2] significantly advanced the field by introducing the CoLES approach based on the contrast between subsequences of different event sequences. However, existing approaches suffer from the non-universality of embeddings: a model either focuses on a current moment to produce embeddings or provides a global picture, blurring important local details [7]. Part of the problem comes from the unavailability of a good local-scale model, as the direct adaptation of, e.g., Contrastive Predictive Coding [26] for event sequences shows inferior results [2]. The complexity here comes from the diverse information available in transactions. A single transaction event can include information about location, merchant category code, timestamp, amount, currency, etc. Straightforward processing here works only if limited information about events available [25].

Our solution approaches both these problems: the lack of a good local representation learning model and the absence of universal, both local and global, properties in a single model. We introduce CMLM (Contrastive Masked Language Model), a local approach that can deal with any input information, and a combination of CMLM with CoLES by proposing a weighted loss function. This approach concludes with the following contributions:

- We propose CMLM based on the masking part of a sequence that can work with a broad range of available input information for each event.
- On top of it, we design a composition of CMLM and CoLES approaches by combining them in a single loss function. This loss is able to capture both local and global properties in a single embedding - and avoid the collapse of representations.
- The developed approach results in an encoder that produces representations with good local and global properties. This effect is evident in multiple financial transactions event sequence datasets.

2 Related work

Self-supervised learning (SSL) is a machine learning paradigm where models are trained using the data itself as labels, making predictions about one part of the data using other parts as a form of supervision [17]. This approach is particularly valuable in leveraging large amounts of unlabeled data, which is often more

readily available than labeled data. The motivation behind SSL arises from the data-hungry nature of deep learning algorithms, which require vast amounts of labeled data to avoid overfitting and biases. By using SSL, models can improve their generalization capabilities and handle out-of-distribution scenarios more effectively. There are two main types of self-supervised frameworks: contrastive and generative [23].

Contrastive methods in self-supervised learning involve learning representations by distinguishing between different instances in the data, often through data augmentation techniques that create pairs of similar and dissimilar examples. These approaches are particularly popular in the computer vision (CV) domain, with prominent examples including SimCLR [10, 11], MoCo [12, 13, 19], and SwAV [9]. Unlike traditional contrastive methods that rely on negative samples, approaches such as BYOL [16], Barlow Twins [34], and VICReg [6] eliminate the need for negative examples, focusing instead on maximizing similarity between different views of the same data instance.

Generative methods, on the other hand, aim to capture the underlying data distribution by generating new data instances or predicting missing parts. In the NLP domain, models like BERT [14, 22, 24] are trained to predict masked tokens using bidirectional context, enhancing their understanding of language structure. Similarly, the GPT [31] models focus on autoregressive tasks, predicting the next token in a sequence. In the CV domain, analogous methods such as BEiT [5] and MAE [20] have emerged, employing mask-based strategies to learn visual representations by predicting occluded parts of an image.

Each framework has its strengths and weaknesses. Previous research has shown that generative approaches excel in capturing local patterns, whereas contrastive methods provide higher-level abstractions beneficial for a comprehensive understanding of the object [30]. These frameworks have been adapted to transactional sequences [2, 7, 25], where global tasks require a holistic understanding of the entire sequence, and local tasks focus on timestamp-specific details. The findings revealed that contrastive methods performed better on global tasks, while generative approaches excelled in addressing local tasks, consistent with previous research in the CV domain.

It is tempting to develop a model that leverages the strengths of both approaches. However, prior research indicates that straightforward multitask training - i.e., simply combining the two losses - is ineffective. As a result, existing work either struggles to integrate the two approaches effectively or resorts to complex methods involving multiple models to achieve this integration [25, 30]. Thus, while both contrastive and generative methods offer distinct advantages, combining them remains a challenging task.

3 Methods

3.1 Problem statement

Self-supervised Learning (SSL). Let $X = \{x_1, ..., x_N\}$ denote an unlabeled set of event sequences. Each sequence $x_i = \{e_{ij}\}_{j=1}^{T_i}$ represents a temporal sequence of

events, where e_{ij} denotes the j^{th} event in the i^{th} sequence, and T_i represents the length of the i^{th} sequence. Each event e_{ij} is characterized by multiple categorical or continuous features. It can include event type as well — and all events are sorted in the sequence according to the occurrence time.

The objective, given X, is to train a neural network $f_\theta : \mathcal{X} \to \mathcal{Z}$, parameterized by θ, to map sequences to representations in a latent space. These representations should effectively capture the temporal dynamics and underlying structure inherent in the sequences. Subsequently, representations acquired through f_θ can be applied across diverse downstream tasks, including classification, regression, and clustering.

This research focuses solely on evaluating the quality of representations for two classification tasks. The evaluation of learned representations involves assessing their performance on two distinct types of downstream tasks [7]. This approach allows for the examination of different aspects of learned embeddings: global, which characterizes the sequence as a whole, and local, which captures the current state of individual instances. We can also fine-tune f_θ for specific tasks or integrate it into larger models to enable multimodal capabilities.

3.2 Model architecture

A neural network model f_θ consists of two main components: an event encoder g_{θ_g} and a sequence encoder s_{θ_s}. The event encoder maps individual events into an embedding space. In this paper, we focus on financial transactions data. For such data modality, each event is characterized by two features: the Merchant Category Code (MCC), the amount and the event time. The MCC, being a categorical variable, is transformed into a k-dimensional embedding for each category. The amount, a continuous variable, is then concatenated with this MCC embedding to form the final embedding of the event. For other event sequences data, we see a similar pattern with a mix of categorical and continuous variables or a single characteristics of event presented. So, our model and methods would benefit any event sequences data processing task.

The full process is illustrated in Figure 1 and follows similar methods described in [3, 33]. Formally, for a sequence $x = \{e_j\}_{j=1}^T$, where e_j represents the j-th event, the event encoder g_{θ_1} maps each event to its corresponding representation r_j:

$$\{e_j\}_{j=1}^T \xrightarrow{g_{\theta_g}} \{r_j\}_{j=1}^T.$$

Subsequently, the sequence encoder s_{θ_s} processes these event representations to generate contextualized representations or hidden states h_j:

$$\{r_j\}_{j=1}^T \xrightarrow{s_{\theta_s}} \{h_j\}_{j=1}^T.$$

In this study, we used a unidirectional single-layer LSTM [21] as the sequence encoder. The performance of this model is similar to transformers for the event sequences data modality, so we opt for a simpler model usage. To represent the entire sequence, we applied an aggregation function to the hidden states.

Specifically, we used the hidden state of the last event as the sequence embedding, as it encapsulates all prior information [7].

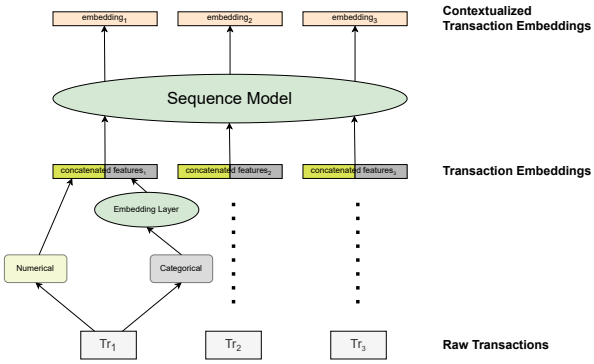

Fig. 1: Diagram of a neural network model showing the transformation from raw transactions to contextualized transaction embeddings.

3.3 Contrastive learning with CoLES

CoLES [2] is a contrastive self-supervised approach. It tries to make positive pairs of subsequences close to each other in the embedding space, while for negative pairs it aims for the opposite effect. There, positive pairs are subsequences of event sequences, while negative pairs are slices from different sequences. Moreover, it uses hard-negative mining [32] to use the most challenging negative pairs from a batch — improving the learning process.

The loss function for CoLES is defined as follows:

$$\mathcal{L}^{\text{CoLES}} = \sum_{x^+ \in P(x)} d^2\left(f_\theta(x), f_\theta(x^+)\right) + \sum_{x^- \in N(x)} \max\left\{0, \rho - d\left(f_\theta(x), f_\theta(x^-)\right)\right\}^2.$$

Here, d is the cosine distance function, $P(x)$ and $N(x)$ represent the sets of positive and negative samples for sample x respectively from a batch, and $\rho > 0$ is a hyperparameter denotes the distance margin for negative pairs.

By making embeddings of different parts of a single sequence close, CoLES tends to produce embeddings that describe sequences as a whole, representing *global properties of an event sequence.*

3.4 Our CMLM

We refer to our approach as Contrastive Masked Language Modeling (CMLM). It combines ideas from Contrastive Predictive Coding [26] (CPC) and Masked Language Modeling [14] (MLM) to learn representations of sequential data.

MLM is a self-supervised learning technique introduced in the NLP domain. A portion of tokens in a sequence is randomly masked with a special token, and the model's objective is to predict the original tokens based on the context provided by the remaining tokens. This prediction task is a classification problem because the tokens are discrete entities that the model must identify from a finite set of possible tokens, so cross-entropy loss is commonly used.

However, MLM cannot be directly applied to transactional sequences because transactions are not discrete tokens; they consist of various categorical and numerical features. To adapt MLM to predict transactions, a common approach in the literature involves training multiple prediction heads, each responsible for predicting a different feature of the transaction [7, 25, 33]. The loss function is then constructed as the sum of cross-entropy losses for categorical features and mean squared errors for numerical features, allowing the model to handle the different types of features appropriately.

While the approach of using multiple prediction heads for transactional data offers a tailored method for handling diverse features, it presents several limitations. One major challenge is the difficulty in capturing the interdependencies between different features. Transactions often involve complex relationships among categorical and numerical features and treating each feature independently may lead to suboptimal modeling of these interactions. Additionally, the approach requires aggregating multiple loss functions — such as cross-entropy for categorical features and mean squared error for numerical features — which can complicate the training process and necessitate careful weighting strategies to avoid bias towards certain types of features.

Instead of employing multiple prediction heads to model each transaction feature individually, we examine an alternative approach inspired by CPC, focusing on predicting transactions in the latent space. By focusing on predicting the embedding of transactions rather than each transaction's individual features, the model aims to capture a compact representation that encompasses the essential information while abstracting away from specific details. For instance, predicting MCCs directly can be challenging due to subtle differences between similar codes. However, such transactions can be mapped close together in the embedding space, allowing the model to understand their shared context without explicitly predicting the MCC, thus simplifying the modeling process.

Integrating concepts from both MLM and CPC, the CMLM approach randomly masks some events in the input sequence with a special token and then forces the model to correctly predict the embeddings of these masked events. So, the loss function is defined as:

$$\mathcal{L}^{\text{CMLM}} = -\sum_{i} \log \frac{e^{\text{sim}(r_i, \hat{r}_i)}}{e^{\text{sim}(r_i, \hat{r}_i)} + \sum\limits_{j \in J} e^{\text{sim}(r_i, \hat{r}_j)}}, \tag{1}$$

where r_i is the embedding of a masked transaction, $\hat{r}_i$ is the predicted embedding of the masked transaction, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, and J is a set of randomly chosen indices serving as negative samples.

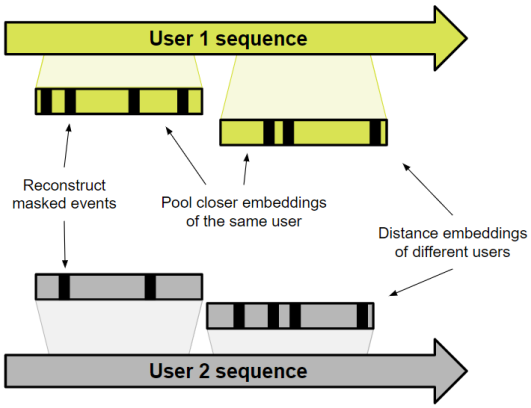

Fig. 2: Scheme of our CoLES+CMLM approach

These works [26, 28] show that minimizing this loss is equivalent to maximizing a lower bound on the mutual information between high-dimensional data and their representations. On the other hand, by construction, produced embeddings should describe *local properties* of the model.

3.5 Hybrid approach: CMLM + CoLES

The approaches described above belong to two different paradigms of self-supervised learning for sequential data: CoLES tends to encode global properties, while our CMLM focuses on local properties.

So, we propose a hybrid approach that combines CMLM's task of predicting masked events in latent space with CoLES's objective of distinguishing between different users in the simplest way possible. The loss function is a straightforward combination of two losses:

$$\mathcal{L}^{\text{CMLM+CoLES}} = \mathcal{L}^{\text{CoLES}} + \lambda \mathcal{L}^{\text{CMLM}}, \tag{2}$$

where the factor λ balances the contributions of CoLES and CMLM to the overall loss. The scheme of our approach is in Figure 2.

4 Experiments

4.1 Datasets

To ensure thorough evaluation, we tested the proposed methods on four public transactional datasets from various data science competitions, following existing benchmarks [2, 7]. Each dataset features transactions with details such as user ID, date, MCC, amount, and other relevant attributes. Descriptive statistics are available in Table 1, with more details provided in the next paragraph.

Table 1: Dataset Statistics

	Churn	**Gender**	**Age**	**DataFusion**
Num. of Transactions	490K	2.9M	26M	8.7M
Num. of Sequences	5K	7.4K	30K	64K
Mean Seq. Length	98.1	388.2	881.7	136.5
Std. Seq. Length	78.1	309.4	124.8	148.9
Num. of Unique MCC	344	184	202	323

Table 2: Model hyperparameters

	Churn	**Gender**	**Age**	**DataFusion**
Embedding size	24	24	24	24
Vocabulary size	344	184	202	323
Hidden size	512	128	512	64
Number of epochs	50	100	50	50

Churn (Rosbank) aims at predicting the probability of customer churn. The binary target labels are nearly balanced. The data spans from October 2016 to April 2018. *Gender (Sber)* involves the prediction of a client's gender based on their transactional activity. It covers the period from January 2022 to April 2023. *Age (Sber)* has the age group label. The multiclass target labels are evenly distributed across four age groups. *DataFusion (VTB)* has the objective to predict a bank customer churn by developing an algorithm that forecasts the likelihood of churn in the subsequent six months. The dates' interval is from October 2021 to March 2023.

To assess both global and local properties of the obtained embeddings, we conducted two distinct downstream tasks for each dataset: sequence classification, as for each dataset we have a corresponding target, and prediction of the next transaction's MCC. For sequence classification, we used CatBoost [29] as the downstream model folllowing [2], as typically results are superior to linear probing and fully connected neural networks. Meanwhile, for the next event type prediction, we trained a linear layer atop of a frozen encoder, following the approach described in [7] — again, here we adopt the best practice available in the literature. We evaluated the performance using the ROC-AUC metric, applying a weighted average for non-binary target variable.

All experiments were conducted using PyTorch [27] and executed on an NVIDIA GeForce RTX 3060 GPU. The datasets were split into training, validation, and test sets with an 80/10/10 ratio to provide a robust evaluation framework. Each experiment was run 5 times to account for variability and enhance the reliability of the results. A summary of the hyperparameters used is provided in Table 2.

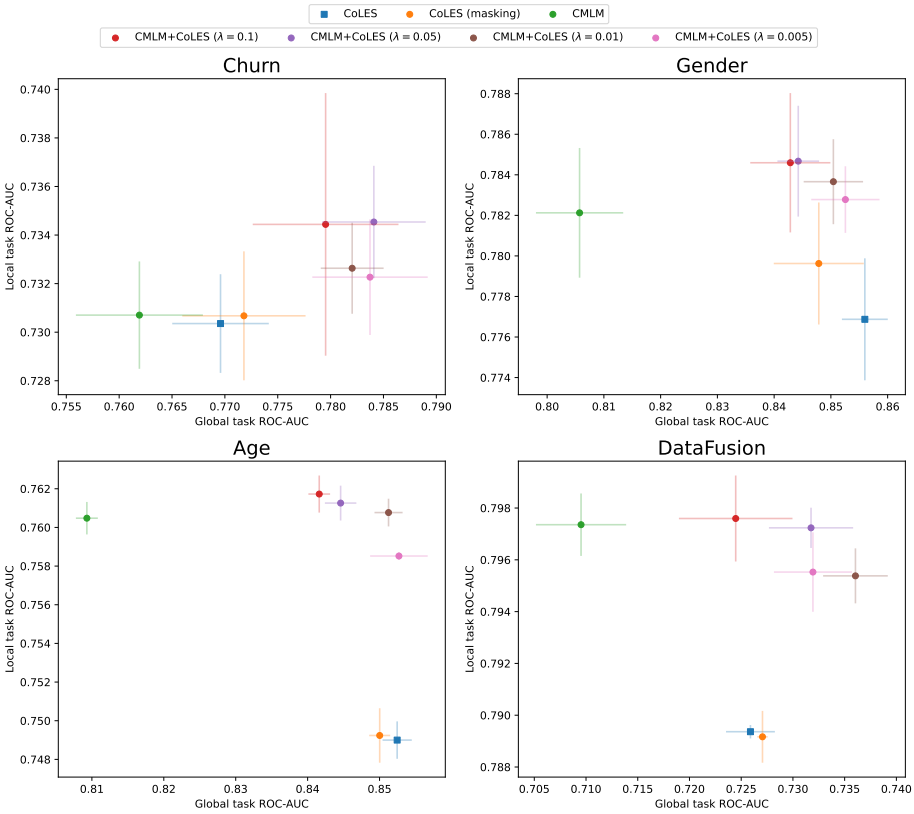

Fig. 3: Quality metrics for global and local embedding tasks, showing ROC-AUC scores across four datasets. Each point represents the mean performance of a model, with error bars indicating standard deviation.

4.2 Results

This study evaluated a hybrid representation learning approach against the baseline method, CoLES, across four datasets: Churn, Gender, Age, and DataFusion. The analysis considered both global and local performance, providing insights into how each method performs relative to CoLES in capturing both overall and timestamp-specific transaction sequence patterns. The results are provided in Table 3 and Figure 3. Additionally, Table 3 compares our methods to two generative approaches - Masked Language Model (MLM) and Autoencoder (AE) - from [7]. For convenience, Table 4 presents the mean performance ranks of the evaluated methods, highlighting our hybrid approach's strong results in both global and local tasks.

Compared to CoLES, CMLM improves in local performance but falls behind when considering global performance. This observation holds for all 4 considered datasets. Thus, CMLM is effective in capturing the current state of individual

Table 3: ROC-AUC metrics for the validation results of global and local properties of embeddings. The results, averaged over five runs, are presented in the format mean ± standard deviation. The best values are **highlighted**, while the second-best values are underlined.

	Churn		**Gender**		**Age**		**DataFusion**	
	Global	Local	Global	Local	Global	Local	Global	Local
CoLES	$0.770_{\pm 0.007}$	$0.730_{\pm 0.003}$	$\mathbf{0.856_{\pm 0.005}}$	$0.777_{\pm 0.004}$	$\underline{0.852_{\pm 0.002}}$	$0.749_{\pm 0.001}$	$0.726_{\pm 0.003}$	$0.789_{\pm 0.000}$
AE	$0.756_{\pm 0.007}$	$0.701_{\pm 0.004}$	$0.676_{\pm 0.021}$	$0.753_{\pm 0.002}$	$0.782_{\pm 0.010}$	$0.722_{\pm 0.007}$	$0.657_{\pm 0.013}$	$0.751_{\pm 0.009}$
MLM	$0.753_{\pm 0.014}$	$0.723_{\pm 0.003}$	$0.833_{\pm 0.004}$	$0.764_{\pm 0.002}$	$0.837_{\pm 0.006}$	$0.714_{\pm 0.002}$	$0.716_{\pm 0.011}$	$0.766_{\pm 0.002}$
CoLES (only masking)	$0.772_{\pm 0.007}$	$0.731_{\pm 0.003}$	$0.848_{\pm 0.009}$	$0.780_{\pm 0.003}$	$0.850_{\pm 0.001}$	$0.749_{\pm 0.001}$	$0.727_{\pm 0.001}$	$0.789_{\pm 0.001}$
CMLM	$0.762_{\pm 0.010}$	$0.731_{\pm 0.004}$	$0.806_{\pm 0.009}$	$0.782_{\pm 0.004}$	$0.809_{\pm 0.002}$	$0.760_{\pm 0.001}$	$0.710_{\pm 0.005}$	$\underline{0.797_{\pm 0.001}}$
CMLM+CoLES ($\lambda = 0.1$)	$0.780_{\pm 0.008}$	$\underline{0.734_{\pm 0.006}}$	$0.843_{\pm 0.007}$	$\mathbf{0.785_{\pm 0.004}}$	$0.842_{\pm 0.002}$	$\mathbf{0.762_{\pm 0.001}}$	$0.724_{\pm 0.005}$	$\mathbf{0.798_{\pm 0.001}}$
CMLM+CoLES ($\lambda = 0.05$)	$\mathbf{0.784_{\pm 0.008}}$	$\mathbf{0.735_{\pm 0.004}}$	$0.844_{\pm 0.004}$	$\mathbf{0.785_{\pm 0.003}}$	$0.845_{\pm 0.002}$	$\underline{0.761_{\pm 0.001}}$	$0.732_{\pm 0.005}$	$\underline{0.797_{\pm 0.001}}$
CMLM+CoLES ($\lambda = 0.01$)	$\underline{0.782_{\pm 0.005}}$	$0.733_{\pm 0.003}$	$0.850_{\pm 0.005}$	$\underline{0.784_{\pm 0.002}}$	$0.851_{\pm 0.002}$	$\underline{0.761_{\pm 0.001}}$	$\mathbf{0.736_{\pm 0.003}}$	$0.795_{\pm 0.001}$
CMLM+CoLES ($\lambda = 0.005$)	$\mathbf{0.784_{\pm 0.009}}$	$0.732_{\pm 0.004}$	$\underline{0.853_{\pm 0.008}}$	$0.783_{\pm 0.002}$	$\mathbf{0.853_{\pm 0.005}}$	$0.759_{\pm 0.000}$	$\underline{0.734_{\pm 0.005}}$	$0.795_{\pm 0.001}$

transactions but struggles with generalizing across entire sequences. This aligns with previous research indicating that generative models excel at local tasks [7].

The proposed hybrid method CMLM+CoLES performed well across several datasets, capturing both global and local transaction patterns. In the Churn dataset, it outperformed CoLES in both global and local tasks. For the Gender dataset, it matched CMLM in local performance while maintaining comparable global performance to CoLES with $\lambda = 0.005$. Similar patterns were observed in the Age and DataFusion datasets. Additional experiments for CoLES with masking shows inferior results compared to this approach. So, the combination of CMLM and CoLES effectively captures both global and local transaction patterns.

The influence of the hyperparameter λ is well-defined: increasing its value places greater emphasis on the CMLM loss, which enhances the model's ability to capture local patterns but reduces its ability to capture global ones. This parameter thus creates a trade-off between the local and global properties of the resulting embeddings.

5 Conclusions

We presented an approach that combines the strengths of contrastive and generative learning methods to improve representation learning for event sequences. The proposed CMLM approach solves a generative task — masked modeling — in the embedding space. This allows the model to focus on the core aspects of the data while ignoring less significant details.

Extensive experiments show, that CMLM outperforms CoLES in local pattern modeling. Furthermore, our hybrid approach CMLM+CoLES, which combines them through a simple multitask learning framework, achieved notable

Table 4: The mean model performance ranks for validation results of the global and local properties of embeddings. All ranks are computed with respect to the ROC-AUC metric and averaged over all datasets. The best values are **highlighted**, while the second-best values are underlined.

	Global task	Local task
CoLES	2.000	3.750
AE	5.750	5.750
MLM	4.500	5.250
CoLES (only masking)	2.500	3.125
CMLM	4.750	1.875
CMLM+CoLES ($\lambda = 0.01$)	**1.500**	**1.250**

improvements in both local and global properties of learned representations. This simplicity contrasts with other complex hybrid methods proposed before. However, one limitation is the need to carefully select the hyperparameter λ, which controls the balance between the generative and contrastive tasks.

In summary, our hybrid approach combining generative and contrastive learning methods significantly advances event sequence modeling. Our universal method enhances both local and global representation learning while maintaining simplicity and flexibility. Future research could extend its application to various sequence types and domains.

6 Acknowledgments

The research was supported by the Russian Science Foundation grant 20-7110135. We also thank our lab team members for providing the code for the baselines.

References

1. Ala'raj, M., Abbod, M.F., Majdalawieh, M., Jum'a, L.: A deep learning model for behavioural credit scoring in banks. Neural Computing and Applications **34**(8), 5839–5866 (2022)
2. Babaev, D., et al.: Coles: Contrastive learning for event sequences with self-supervision. In: SIGMOD. pp. 1190–1199 (2022)
3. Babaev, D., et al.: E.T.-RNN: Applying deep learning to credit loan applications. In: ACM SIGKDD. p. 2183–2190. KDD '19, Association for Computing Machinery, New York, NY, USA (2019)
4. Bany Mohammed, A., Al-Okaily, M., Qasim, D., Khalaf Al-Majali, M.: Towards an understanding of business intelligence and analytics usage: Evidence from the banking industry. International Journal of Information Management Data Insights **4**(1), 100215 (2024)
5. Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT pre-training of image transformers. In: ICLR (2022)

6. Bardes, A., Ponce, J., Lecun, Y.: Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In: ICLR (2022)
7. Bazarova, A., et al.: Universal representations for financial transactional data: embracing local, global, and external contexts (2024)
8. Bueno, L.A., et al.: Impacts of digitization on operational efficiency in the banking sector: Thematic analysis and research agenda proposal. International Journal of Information Management Data Insights **4**(1), 100230 (2024)
9. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. NeurIPS **33**, 9912–9924 (2020)
10. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
11. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. NeurIPS **33**, 22243–22255 (2020)
12. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning (2020)
13. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: CVPR. pp. 9640–9649 (2021)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Fursov, I., et al.: Adversarial attacks on deep models for financial transaction records. In: ACM SIGKDD. pp. 2868–2878 (2021)
16. Grill, J.B., et al.: Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS **33**, 21271–21284 (2020)
17. Gui, J., et al.: A survey on self-supervised learning: Algorithms, applications, and future trends. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
18. Hassani, H., et al.: Deep learning and implementations in banking. Annals of Data Science **7**, 433–446 (2020)
19. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020)
20. He, K., et al.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
22. Lan, Z., et al.: ALBERT: A lite BERT for self-supervised learning of language representations. In: ICLR (2019)
23. Liu, X., et al.: Self-supervised learning: Generative or contrastive. IEEE transactions on knowledge and data engineering **35**(1), 857–876 (2021)
24. Liu, Y., et al.: Roberta: A robustly optimized bert pretraining approach (2019)
25. Moskvoretskii, V., et al.: MLEM: Generative and contrastive learning as distinct modalities for event sequences (2024), https://arxiv.org/abs/2401.15935
26. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
27. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. NeurIPS **32** (2019)
28. Poole, B., et al.: On variational bounds of mutual information. In: ICML. pp. 5171–5180. PMLR (2019)
29. Prokhorenkova, L., et al.: Catboost: unbiased boosting with categorical features. In: NeurIPS. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)

30. Qi, Z., et al.: Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In: ICML. pp. 28223–28243. PMLR (2023)
31. Radford, A., et al.: Improving language understanding by generative pre-training (2018)
32. Robinson, J., Chuang, C.Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. In: ICLR (2021)
33. Udovichenko, I., et al.: SeqNAS: Neural architecture search for event sequence classification. IEEE Access **12**, 3898–3909 (2024)
34. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: ICML. pp. 12310–12320. PMLR (2021)